

REMARKS

Claims 1-41 are pending in the application. No claims are presently allowed.

Claims 40 and 41 have been amended to change "processor slice" to "processing slice" for consistency with the other claims.

Claim 41 has been amended to correct a clerical error by changing "units" to "unit."

Claim Objections

The Examiner objected to claim 41 as informal for reciting "at least one peripheral units." This has been corrected.

Claim Rejections § 112

The Examiner rejected claims 1-41 under 35 U.S.C. § 112, first paragraph for lack of enablement, stating that the specification does not support the claim limitation that the processing slice is capable of executing the instructions from more than one of the plurality of threads concurrently in a clock cycle. The Examiner cited the Specification for the statement that the "processing slice operates by interleaving the execution of instructions from the four threads," and stated the interleave technique does not enable the processing slice to execute multiple instructions simultaneously.

The remainder of the cited sentence in the Specification states that the processing slice includes "the ability to execute several instructions concurrently in the same clock cycle." Page 8, lines 24-25. Thus, the use of the term "interleaving" was not meant to exclude this ability. The Specification also explains that within each cycle, "one or more instructions may selected for execution concurrently." Page 14, lines 12-13. A flowchart of this process is shown in Fig. 6. A single cycle begins in block 610 and ends in block 640. Between the start and end blocks 630_i through 630_k show multiple instructions executed concurrently. These portions of the Specification are sufficient to enable the claim limitation.

Further, a person skilled in the art would be able to produce the invention with this limitation without under experimentation. The attached article, Tullsen et al, "Simultaneous Multithreading: Maximizing On-Chip Parallelism," Proc. 22nd Annual International Symposium on Computer Architecture, 1995, 392-403, was available in the prior art before the filing date of the application. The article describes simulations of simultaneous multithreading. In this

method, the processor has multiple issue slots in each cycle, and instructions from multiple threads can fill the slots in each cycle (section 4.1). The knowledge of a person skilled in the art would be enabling for the invention.

Additional enablement is found in US Provisional Application No. 60/166,686, of which this application claims the benefit. The provisional application describes an instruction buffer that can hold up to four instruction words for each active thread. In each clock cycle, up to four words are read from the instruction buffer, each containing two instruction elements from a distinct thread. A subset of these eight instruction elements is presented to an instruction window. The instruction window has six or eight window slots that can hold one instruction element. The size of the subset is limited by the number of instruction window slots that will become free on the next clock cycle. An instruction decode logic block identifies complete instructions in the instruction window and assigns complete instructions to functional units.

The provisional application was not incorporated by reference, however the USPTO has proposed a new rule (37 C.F.R. § 1.57) that would allow for incorporation of matter from a prior application (1275 O.G. 23, 10/07/2003). The rule is scheduled to go into effect on 05/28/2004, which is within the statutory period for reply. If the rule goes into effect, this material can be incorporated if necessary to provide additional enablement.

It appears that this limitation was not considered by the Examiner in making a 35 U.S.C. § 103 rejection. In the event that the enablement rejection is withdrawn, Applicant respectfully requests that the finality of the obviousness rejection be withdrawn and the obviousness rejection reconsidered

Claim Rejections § 103

The Examiner rejected claims 1-12, 14-25, 27-38, and 40-41 under 35 U.S.C. § 103 as unpatentable over the combination of Bucher (US 5,421,014) and Motomura (US 5,814,727).

Claim 1 is to an apparatus comprising: a peripheral bus coupled to a peripheral unit and a processing slice coupled to the peripheral bus. The peripheral bus transfers peripheral information including a command message specifying a peripheral operation. The processing slice executes a plurality of threads comprising instructions. The threads include a first thread sending the command message to the peripheral unit. The processing slice comprises a functional unit to perform a register operation specified in the instructions in each thread. The

processing slice executes the instructions from more than one of the plurality of threads concurrently in a clock cycle.

Bucher discloses a software architecture for implementing multi-thread control of a peripheral interface, especially a SCSI interface. The software operates at the driver level and manages multiple peripheral requests. A higher level program sends a peripheral request to the driver. When the operation is complete, the driver sends the result to the high level program. The high level program can continue its own execution without waiting for the result from the peripheral. (Col. 3, lines 22-35 and 52-63).

Motomura discloses a multi-thread parallel processor system having a plurality of processors and an ordered multithread executing system (Fig. 1). The ordered multithread executing system determines which thread will execute on each processor.

In order to make a *prima facie* case of obviousness, the references must disclose each limitation of the claims. Bucher does not disclose the processing slice recited in claim 1. The processing slice is able to process several threads simultaneously, that is, instructions from two or more threads may be executing on the slice hardware at the same time. This is not the case with Bucher, which assumes a conventional processor in which instructions from just one thread are being executed at any time, and generally it takes several instruction executions to switch the processor from executing one thread to executing another thread. Claim 1 is not anticipated by Bucher. Claims 14, 27, and 40 also recite the processing slice and are also not anticipated by Bucher.

As in Bucher, Motomura does not disclose a processing slice as in claim 1. The processing slice contains a functional unit to perform a register operation specified in the instructions in each of the plurality of threads. Thus, the functional unit is shared among multiple, simultaneously executing threads. Motomura does not disclose details of the processor, but it is known that a processor usually contains a functional unit for performing register operations. In Motomura, each executing thread has its own processor including a functional unit which does not perform operations from any other thread. Thus, each functional unit would have a significant amount of idle time. In the present invention, the functional unit is shared among the threads and is used more efficiently.

The Examiner stated that each of Motomura's processors is equivalent to Applicant's processing slice. However, Motomura's processors are not able to execute instructions from

more than one thread at a time. The processor is dependent upon the ordered multithread executing system to change execution from one thread to another (col. 8, lines 13-51). The ordered multithread executing system does not start execution of a thread on a processor unless that processor had been executing a thread that has completed or gone into a waiting state. When the new thread is assigned to the processor, the processor stops executing the former thread. Although the processor may execute more than one thread over the course of an entire program, at any given time, only one thread is executing on the processor. The functional unit of that processor can only be used by the one thread that is executing.

The processing slice of the present invention can execute more than one thread simultaneously. This differs from both the processor and the parallel processing system of Motomura. In the present invention, the processing slice is able to dispatch an instruction from any currently executing thread to any of the functional units within the processing slice. This is more efficient than the processing system of Motomura in that fewer functional units are needed. It is unlikely that all currently executing threads would need constant use of the functional units. Since each thread can use any functional unit, there can be full utilization of functional units without any delay to wait for a functional unit to be available. This also reduces the number of lost clock cycles that are not fully utilized. In the system of Motomura, the functional units are not shared among threads. Each currently executing thread has a dedicated functional unit which may not be completely utilized.

Claims 2-12, 15-25, 28-38, and 41 depend from and contain all the limitations of claims 1, 14, 27, or 41 and are asserted to differ from the references in the same way as claims 1, 14, 27, and 41.

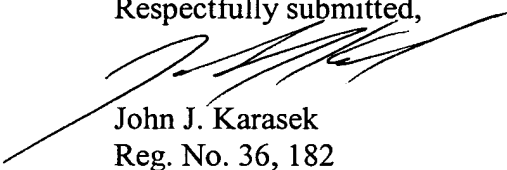
The Examiner rejected claims 13, 26, and 39 under 35 U.S.C. § 103 as unpatentable over the combination of Bucher, Motomura, and Hiraoka (US 5,418,917).

Hiraoka discloses a method and apparatus for controlling a conditional branch instruction in a pipeline type data processing apparatus. As in Bucher and Motomura, Hiraoka does not disclose a processing slice as recited in claims 1 (13 dependent thereon), 14 (26 dependent thereon), and 27 (39 dependent thereon). Hiraoka discloses the execution of only one instruction at a time. As none of the references discloses the processing slice, there is no *prima facie* case of obviousness.

In view of the foregoing, it is submitted that the application is now in condition for allowance.

In the event that a fee is required, please charge the fee to Deposit Account No. 50-0281, and in the event that there is a credit due, please credit Deposit Account No. 50-0281.

Respectfully submitted,



John J. Karasek
Reg. No. 36, 182
Phone No. 202-404-1552
Associate Counsel (Patents)
Naval Research Laboratory
4555 Overlook Ave, SW
Washington, DC 20375-5325

Prepared by:
Joseph T. Grunkemeyer
Reg. No. 46,746
Phone No. 202-404-1556